Refine Search

Search Results -

Terms	Documents
L4 AND translate OR translator	11070

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L5

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Monday, December 13, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L5	L4 AND translate OR translator	11070	L5
L4	L3 AND ((choose OR select) OR ((Click-on OR click ADJ ON)))	1478	L4
L3	L ANd (dynamic ADJ memory)	2373	L3
L2	L1 ANd design AND test ANd graphical and translator	3	L2
L1	717/124.ccls.	269	L1

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 3 of 3 returned.

☐ 1. Document ID: US 6721941 B1

L2: Entry 1 of 3 File: USPT Apr 13, 2004

US-PAT-NO: 6721941
DOCUMENT-IDENTIFIER: US 6721941 B1

TITLE: Collection of timing and coverage data through a debugging interface

DATE-ISSUED: April 13, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Morshed; Farokh	Amherst	NH		
Meagher; Robert	Milford	NH		

US-CL-CURRENT: 717/127; 709/217, 714/38, 717/124, 717/126, 717/128, 717/129, 717/130, 717/131, 719/328

ABSTRACT:

Techniques for gathering execution information about an application, such as a distributed application, are described. Key communication points in cross execution context calls, such as remote procedure calls, are determined and control is transferred to instrumentation routines to insert and extract execution information. Outgoing remote procedure calls are intercepted on a client that inserts call origin information into the request sent to a server system. Messages received by a server are intercepted. The server system extracts the call origin information and additionally inserts other information in a response sent to the client system upon completion of a remote procedure call. In turn, the client system intercepts the response and extracts other performance information. On each client and server system, information is gathered by a reader and forwarded to a local collector. This information may be further forwarded to and correlated by a client collector from one or more remote server collectors in accordance with processes of each distributed application. Various statistics for a distributed application may be determined in addition to per process statistics. These include wire time, code coverage as related to the distributed application, remote procedure call tracing, and performance profiling. A variety of techniques are described to obtain program execution information in connection with an executing application including instrumentation techniques and use of a debugger interface to obtain profiling and other execution information. All of the program execution data may be collected and correlated at one or more particular points using other techniques described to represent coordinated application monitoring.

50 Claims, 82 Drawing figures
Exemplary Claim Number: 1

Number of Drawing Sheets: 77

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 2. Document ID: US 6701514 B1

L2: Entry 2 of 3 File: USPT Mar 2, 2004

US-PAT-NO: 6701514
DOCUMENT-IDENTIFIER: US 6701514 B1

TITLE: System, method, and article of manufacture for test maintenance in an automated scripting framework

DATE-ISSUED: March 2, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Haswell; John Jeffrey	Herndon	VA		
Young; Robert J.	Charlestown	MA		
Schramm; Kevin	Rose Valley	PA		

US-CL-CURRENT: 717/115; 707/102, 717/124

ABSTRACT:

A system, method and article of manufacture are provided for affording test maintenance in an automated scripting framework. First, a plurality of test scripts are developed. Then, the plurality of test scripts are stored in a centrally located database. A user is then allowed to edit a specific test script located on the centrally located database. Finally, the user edits to the specific test script are propagated to each of the plurality of test scripts.

18 Claims, 82 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 52

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw D

☐ 3. Document ID: US 6412106 B1

L2: Entry 3 of 3 File: USPT Jun 25, 2002

US-PAT-NO: 6412106
DOCUMENT-IDENTIFIER: US 6412106 B1

TITLE: Graphical system and method for debugging computer programs

DATE-ISSUED: June 25, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Leask; Gary M.	Dallas	TX		
Huffman; Dale L.	Allen	TX		

US-CL-CURRENT: 717/124; 717/125, 717/148

ABSTRACT:

A system and method for graphically debugging a computer program is disclosed. In a preferred embodiment, a graphical debugging environment is provided, which is capable of displaying a graphical representation of an application program to be debugged. Thereafter, the graphical debugging environment allows a user to insert debugging tools, such as breakpoints, directly into the graphical representation of the application program. Thus, a user is not required to interact with the textual source code of an application program when debugging it. The graphical debugging environment may display indicators illustrating where debug tools have been inserted within the application program. In a preferred embodiment, the graphical debugging environment allows a user to perform debugging during an application program's runtime. Thus, a user is not required to halt an application program prior to debugging it. Also, in a preferred embodiment the graphical debugging environment executing on a local computer may be used to debug an application program residing on a remote computer.

52 Claims, 9 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 7

Terms	Documents
L1 ANd design AND test ANd graphical and translator	3

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 6 of 6 returned.

☐ 1. Document ID: US 6789252 B1

L6: Entry 1 of 6 File: USPT Sep 7, 2004

US-PAT-NO: 6789252
DOCUMENT-IDENTIFIER: US 6789252 B1

TITLE: Building business objects and business software applications using dynamic object definitions of ingrediential objects

DATE-ISSUED: September 7, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Burke; Miles D.	Phoenix	AZ	85016	
Solar, Jr.; Richard J.	Phoenix	AZ	85018	

US-CL-CURRENT: 717/100; 717/103

ABSTRACT:

A method and system are provided for providing an open and extensible object definition framework that manages business object definitions as specifications. This framework may be used to dynamically define any object that is to be processed by a computer. Objects can include Properties, Classifications, Knowledge, Business Objects, and Business Rules to name a few. Some examples of typical Business Objects include: business and social entities; locations, including spaces, places and channels; activities, including events and processes; items, including products and services; and business records, including orders and other forms of demand, inventory, jobs, deliverables, statements, transaction history et. al. The method and system may be used to define any object that is to be processed by a computer. Objects can include Properties, Classifications, Knowledge, Business Objects, and Business Rules to name a few. Typical Business Objects include: Business and social entities; Locations including spaces, places, and channels; Activity including events and processes; Items including products and services; Business Records including orders and other forms of demand, inventory, jobs, deliverables, statements, transaction history et. al.

237 Claims, 127 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 72

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

☐ 2. Document ID: US 6772408 B1

L6: Entry 2 of 6 File: USPT Aug 3, 2004

US-PAT-NO: 6772408
DOCUMENT-IDENTIFIER: US 6772408 B1

TITLE: Event model using fixed-format text strings to express event actions

DATE-ISSUED: August 3, 2004

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Velonis; John	Dobbs Ferry	NY		
Deb Nath; Rituraj	Stamford	CT		

US-CL-CURRENT: 717/100; 709/223, 709/224, 715/763, 717/101, 717/102, 717/103, 717/106, 717/108, 717/109

ABSTRACT:

The present invention relates to computer software for initiating actions in response to user input, e.g., on an Internet web page. More particularly, the invention provides an improved event model comprising software components (designated "Fidgets" herein) which use fixed-format text strings to express event actions in a content delivery subsystem. In particular, the present invention provides a method for setting the event target property for a user event without the need to recompile software.

16 Claims, 3 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 3

☐ 3. Document ID: US 6618851 B1

L6: Entry 3 of 6 File: USPT Sep 9, 2003

US-PAT-NO: 6618851
DOCUMENT-IDENTIFIER: US 6618851 B1

TITLE: Method and apparatus for state-reversion

DATE-ISSUED: September 9, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Zundel; Robert E.	Wilsonville	OR		
Mullin; Doug	Portland	OR		
Synge; James	Portland	OR		
Borduin; Scott	Lake Oswego	OR		

US-CL-CURRENT: 717/103; 711/156, 717/120

ABSTRACT:

A programming environment can be modified to provide automatic support for reverting program memory states. Such memory reversions are used to provide automatic support for state-reversion, undo, redo, and abort operations for application programs written with the programming environment. Memory allocation code (e.g., functions, procedures, etc.) are modified to mark allocated memory as protected, and an exception handler is assigned to such memory. Attempts to access the memory cause an exception to be generated. This exception is caught, providing opportunity for the memory to be preserved before it is modified. Previous memory states can be retrieved by restoring such stored memory states. State-reversion can be effected by creating a new current memory state corresponding to a previous (e.g., retrieved) memory state.

16 Claims, 7 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 6

☐ 4. Document ID: US 6601233 B1

L6: Entry 4 of 6 File: USPT Jul 29, 2003

US-PAT-NO: 6601233
DOCUMENT-IDENTIFIER: US 6601233 B1

TITLE: Business components framework

DATE-ISSUED: July 29, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Underwood; Roy Aaron	Long Grove	IL		

US-CL-CURRENT: 717/102; 717/100, 717/101, 717/103, 717/104, 717/106, 717/107

ABSTRACT:

A method of generating software based on business components. A plurality of logical business components in a business are first defined with each business component having a plurality of capabilities. Next, functional interrelationships are identified between the logical business components. Code modules are then generated to carry out the capabilities of the logical business components and the functional interrelationships between the logical business components, wherein the code modules represent a transformation of the logical business components to their physical implementation, while ensuring the capabilities that are carried out by each code module are essentially unique to the logical business component associated with the code module. Next, the functional aspects of the code modules and the functional relationships of the code modules are tested. The code modules are then subsequently deployed in an e-commerce environment.

18 Claims, 177 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 111

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 5. Document ID: US 5960200 A

L6: Entry 5 of 6 File: USPT Sep 28, 1999

US-PAT-NO: 5960200
DOCUMENT-IDENTIFIER: US 5960200 A

TITLE: System to transition an enterprise to a distributed infrastructure

DATE-ISSUED: September 28, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Eager; Timothy	Fullerton	CA		
Anand; Madhav	Cambridge	MA		
Aslanian; Edouard	Hermosa Beach	CA		

US-CL-CURRENT: 717/147; 703/13, 703/20, 705/7, 709/201, 717/103, 717/104, 717/108

ABSTRACT:

An automated system transitions an entire enterprise to a distributed infrastructure. The system includes a process for organizing and managing the transition, a multi-tiered client/server architecture that adheres to open systems standards, a system to automate the transition of existing applications to this architecture, and a system to enable the creation or modification of applications based on this architecture.

54 Claims, 36 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 36

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. D

☐ 6. Document ID: US 5872932 A

L6: Entry 6 of 6 File: USPT Feb 16, 1999

US-PAT-NO: 5872932
DOCUMENT-IDENTIFIER: US 5872932 A

TITLE: Persistence specification system and method for producing persistent and transient submaps in a management station for a data communication network

DATE-ISSUED: February 16, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Schettler; Robert Dwight	Fort Collins	CO		
McCollom; William Girard	Fort Collins	CO		
Haimson; David M.	Fort Collins	CO		

US-CL-CURRENT: 709/226; 717/103

ABSTRACT:

Discovery/layout software configures a general purpose computer system to act as a management station using an industry standard SNMP protocol. The discovery/layout software has a discovery mechanism and a layout mechanism which, in combination, permit the discovery/layout software to provide various submaps to a display for illustrating network topology, which includes devices and device interconnections of the network. The submaps correspond to various hierarchical views of the network. Significantly, a persistence specification mechanism is provided in the discovery/layout software for specifying a submap as either transient (generated upon demand) or persistent (exists whether demanded or not). An integrating application as well as the user can identify a submap as persistent. This feature enables better interfacing of the integrating application with the station, thereby providing more information to the user. This feature further minimizes memory requirements as well as requisite processing time due to the elimination of unnecessary submaps and the elimination of processing of topology changes relative to the unnecessary submaps.

18 Claims, 19 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 18

Terms	Documents
(717/103).ccls. AND design ANd test and (graphical)	6